

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method, comprising:

determining an identifier for dynamically loadable code;

pushing the identifier onto a unidirectional communication link;

determining whether an availability schedule for the dynamically loadable code;

pushing the availability schedule onto the unidirectional communication link; and

pushing the dynamically loadable code onto the unidirectional communication link according to the availability schedule, wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.
2. (Original) The method of claim 1, further comprising:

wherein the dynamically loadable code comprises a class definition with an object oriented programming language; and

wherein the identifier identifies the class definition.
3. (Original) The method of claim 1, wherein the unidirectional communication link is a selected one of: a television data transmission, an MPEG-2 transport stream, and IP-multicast.

4. (Original) The method of claim 1, further comprising:

receiving data over the unidirectional communication link with a selected one of: a set top box, a personal digital assistant, a portable computer, a handheld computer, and a wireless appliance.

5. (Original) The method of claim 1, further comprising:

receiving the identifier and the availability schedule over the unidirectional communication link; and retrieving the dynamically loadable code from said communication link according to the availability schedule.

6. (Original) The method of claim 5, further comprising:

determining whether the dynamically loadable code is required for executing an application program; and
performing said retrieving responsive to said determining.

7. (Original) An apparatus, comprising:

a machine accessible medium providing instructions, which when executed by a machine, are capable of directing the machine to perform the operations of claim 1.

8. (Original) The apparatus of claim 7, said instructions including further instructions to direct the machine to perform the operations of claim 2.

9. (Original) The apparatus of claim 7, wherein the unidirectional communication link is a selected one of: a television data transmission, an MPEG-2 transport stream, and IP-multicast.

10. (Original) The apparatus of claim 7, said instructions including further instructions to direct the machine to perform the operations of claim 4.

11. (Original) The apparatus of claim 7, said instructions including further instructions to direct the machine to perform the operations of claim 5.

12. (Original) The apparatus of claim 11, said instructions including further instructions to direct the machine to perform the operations of claim 6.

13. (Currently Amended) A method, comprising:

preparing a manifest for dynamically loadable code, said manifest comprising an identifier for dynamically loadable code, and an availability schedule; pushing the manifest onto a unidirectional communication link; and

pushing the dynamically loadable code onto said communication link according to the availability schedule, wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.

14. (Original) The method of claim 13, further comprising:

wherein the dynamically loadable code comprises a class definition in an object oriented programming language; and
wherein the identifier identifies the class definition.

15. (Original) The method of claim 13, wherein the dynamically loadable code is written in a selected one of: Java, Objective-C, C++, SmallTalk, Modula-3, Component Object Model, and an object-oriented scripting language.

16. (Original) The method of claim 13, wherein the unidirectional communication link is a selected one of: a television data transmission, an MPEG-2 transport stream, and IP-multicast.

17. (Original) The method of claim 13, further comprising:
receiving the unidirectional communication link with a selected one of: a set top box, a personal digital assistant, a portable computer, a handheld computer, and a wireless appliance.

18. (Original) The method of claim 13, further comprising:
receiving the manifest over said communication link;
recording the identifier and the availability schedule; and
retrieving the dynamically loadable code when it is pushed onto said communication link according to the availability schedule.

19. (Original) The method of claim 14, further comprising:

determining whether the dynamically loadable code is required for executing an application program; and
performing said retrieving responsive to said determining.

20. (Currently Amended) A method for mirroring a Java-type archive file, comprising:

preparing a manifest for a Java-type archive file, said manifest comprising identifiers for objects of the Java-type archive file, and an availability schedule for said objects;

pushing the manifest onto a unidirectional communication link; and

pushing said objects of the Java-type archive file onto the unidirectional communication link in accordance with the availability schedule, wherein the Java-type archive file is to modify the functionality of executing program code without stopping the executing program code.

21. (Original) The method of claim 20, further comprising:

executing programming code;

determining whether an unavailable object is required for said executing;

determining whether the manifest includes an identifier corresponding to the object; and

receiving said required object over the unidirectional communication link.

22. (Original) The method of claim 21, further comprising:

storing said received object in a temporary memory location disposed within a device;
wherein resetting the device causes said received object to be discarded.

23. (Original) The method of claim 20, wherein the manifest for the Java-type archive file includes purchasing data for said objects of the Java-type archive file, the method further comprising: identifying an unavailable object that is required for executing a program; determining whether the manifest includes an identifier corresponding to the object; receiving said required object over the unidirectional communication link; and purchasing said required object in accord with said purchasing data.

24. (Currently Amended) A method for obtaining dynamically loadable code over a push-only network, comprising: receiving, over the push-only network, a manifest for dynamically loadable code, said manifest comprising an identifier for dynamically loadable code, and an availability schedule; and receiving, over the push-only network, the dynamically loadable code in accord with the availability schedule, wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.

25. (Original) The method of claim 24, wherein the dynamically loadable code comprises a selected one of: a single object oriented object, a plurality of object oriented object definitions, and a Dynamic Link Library (DLL).

26. (Original) The method of claim 24, further comprising:
determining whether an application program requires dynamically loadable code; and
determining whether the manifest includes an identifier corresponding to said
dynamically loadable code.

27. (Original) The method of claim 26, further comprising:
inspecting a CLASSPATH environment for a class containing said required dynamically
loadable code; and
determining whether said required dynamically loadable code is unavailable.

28. (Original) The method of claim 27, further comprising:
adding said received dynamically loadable code to the CLASSPATH environment.

29. (Original) The method of claim 24, wherein the dynamically loadable code
comprises a Java-type programming language class, the method further comprising:
inspecting a CLASSPATH environment for a class containing the dynamically loadable
code; and determining whether said required dynamically loadable code is unavailable,
and responsive thereto, performing said receiving the dynamically loadable code.

30. (Original) The method of claim 24, further comprising:
adding said received dynamically loadable code to a local storage for dynamically
loadable code.

31. (Currently Amended) An apparatus, comprising:

a machine accessible medium providing instructions, which when executed by a machine, are capable of directing the machine to perform:

preparing a manifest for dynamically loadable code, said manifest comprising an identifier for dynamically loadable code, and an availability schedule;

pushing the manifest onto a unidirectional communication link; and

pushing the dynamically loadable code onto said communication link according to the availability schedule, wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.

32. (Original) The apparatus of claim 31, said instructions including further instructions to direct the machine to perform:

receiving the unidirectional communication link with a selected one of: a set top box, a personal digital assistant, a portable computer, a handheld computer, and a wireless appliance.

33. (Original) The apparatus of claim 31, said instructions including further instructions to direct the machine to perform:

receiving the manifest over said communication link;

recording the identifier and the availability schedule; and

retrieving the dynamically loadable code when it is pushed onto said communication link according to the availability schedule.

34. (Original) The apparatus of claim 31, said instructions including further instructions to direct the machine to perform:

determining whether the dynamically loadable code is required for executing an application program; and performing said retrieving responsive to said determining.

35. (Currently Amended) An apparatus for mirroring a Java-type archive file, comprising:

a machine accessible medium providing instructions, which when executed by a machine, are capable of directing the machine to perform:

preparing a manifest for a Java-type archive file, said manifest comprising identifiers for objects of the Java-type archive file, and an availability schedule for said objects;

pushing the manifest onto a unidirectional communication link; and

pushing said objects of the Java-type archive file onto the unidirectional communication link in accordance with the availability schedule, wherein the Java-type archive file is to modify the functionality of executing program code without stopping the executing program code.

36. (Original) The apparatus of claim 35, said instructions including further instructions to direct the machine to perform:

determining whether an unavailable object is required for executing an application;

determining whether the manifest includes an identifier corresponding to the object; and

receiving said required object over the unidirectional communication link.

37. (Original) The apparatus of claim 35, said instructions including further instructions to direct the machine to perform:
determining whether an unavailable object is required for executing an application;
determining whether the manifest includes an identifier corresponding to the object; and
receiving said required object over the unidirectional communication link.

38. (Original) The apparatus of claim 35, said instructions including further instructions to direct the machine to perform:
including purchasing data for said objects of the Java-type archive file in the manifest; identifying an unavailable object that is required for executing a program;
determining whether the manifest includes an identifier corresponding to the object;
receiving said required object over the unidirectional communication link; and
purchasing said required object in accord with said purchasing data.

39. (Currently Amended) An apparatus for obtaining dynamically loadable code over a push-only network, comprising:
a machine accessible medium providing instructions, which when executed by a machine, are capable of directing the machine to perform: receiving, over the push-only network, a manifest for dynamically loadable code, said manifest comprising an identifier for dynamically loadable code, and an availability schedule; and receiving, over the push-only network, the dynamically loadable code in accord with the availability schedule,

wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.

40. (Original) The apparatus of claim 39, said instructions including further instructions to direct the machine to perform:

determining whether an application program requires dynamically loadable code; and

determining whether the manifest includes an identifier corresponding to said dynamically loadable code.

41. (Original) The apparatus of claim 40, said instructions including further instructions to direct the machine to perform:

inspecting a CLASSPATH environment for a class containing said required dynamically loadable code; and

determining whether said required dynamically loadable code is unavailable.

42. (Original) The apparatus of claim 40, said instructions including further instructions to direct the machine to perform:

adding said received dynamically loadable code to the CLASSPATH environment.

43. (Original) The apparatus of claim 39, said instructions including further instructions to direct the machine to perform:

inspecting a CLASSPATH environment for a class containing the dynamically loadable code; and

determining whether said required dynamically loadable code is unavailable, and responsive thereto, performing said receiving the dynamically loadable code.

44. (Currently Amended) A system, comprising:

at least one processor; and

a readable medium having instructions encoded thereon, which when executed by the processor, are capable of directing the processor to perform:

preparing a manifest for dynamically loadable code, said manifest comprising an identifier for dynamically loadable code, and an availability schedule;

pushing the manifest onto a unidirectional communication link; and

pushing the dynamically loadable code onto said communication link according to the availability schedule, wherein the dynamically loadable code is to modify the functionality of executing program code without stopping the executing program code.

45. (Original) The system of claim 44, said instructions including further instructions to direct the processor to perform:

receiving the unidirectional communication link with a selected one of: a set top box, a personal digital assistant, a portable computer, a handheld computer, and a wireless appliance.

46. (Original) The system of claim 44, said instructions including further instructions to direct the processor to perform:

receiving the manifest over said communication link;

recording the identifier and the availability schedule; and
retrieving the dynamically loadable code when it is pushed onto said communication link according to the availability schedule.

47. (Currently Amended) A system for mirroring a Java-type archive file, comprising:
at least one processor; and
a readable medium having instructions encoded thereon, which when executed by the processor, are capable of directing the processor to perform:
preparing a manifest for a Java-type archive file, said manifest comprising identifiers for objects of the Java-type archive file, and an availability schedule for said objects;
pushing the manifest onto a unidirectional communication link; and
pushing said objects of the Java-type archive file onto the unidirectional communication link in accordance with the availability schedule, wherein the Java-type archive file is to modify the functionality of executing program code without stopping the executing program code.

48. (Original) The system of claim 47, said instructions including further instructions to direct the processor to perform:
determining whether an unavailable object is required for executing an application;
determining whether the manifest includes an identifier corresponding to the object; and
receiving said required object over the unidirectional communication link.

49. (Original) The system of claim 47, said instructions including further instructions to direct the processor to perform:

including purchasing data for said objects of the Java-type archive file in the manifest;
identifying an unavailable object that is required for executing a program;
determining whether the manifest includes an identifier corresponding to the object;
receiving said required object over the unidirectional communication link; and
purchasing said required object in accord with said purchasing data.

50. (Currently Amended) A system for obtaining dynamically loadable code over a push-only network, comprising:

at least one processor; and a readable medium having instructions encoded thereon,
which when executed by the processor, are capable of directing the processor to perform:

receiving, over the push-only network, a manifest for dynamically loadable code,
said manifest comprising an identifier for dynamically loadable code, and an availability
schedule; and

receiving, over the push-only network, the dynamically loadable code in accord
with the availability schedule, wherein the dynamically loadable code is to modify the
functionality of executing program code without stopping the executing program code.

51. (Original) The system of claim 50, said instructions including further instructions to direct the processor to perform:

determining whether an application program requires dynamically loadable code; and

determining whether the manifest includes an identifier corresponding to said dynamically loadable code.

52. (Original) The system of claim 50, said instructions including further instructions to direct the processor to perform:

inspecting a CLASSPATH environment for a class containing said required dynamically loadable code;

determining whether said required dynamically loadable code is unavailable; and

adding said received dynamically loadable code to the CLASSPATH environment.